ALICE Grid Services

Federico Carminati (CERN/ALICE)

FNAL, March 24, 2005

Disclaimer

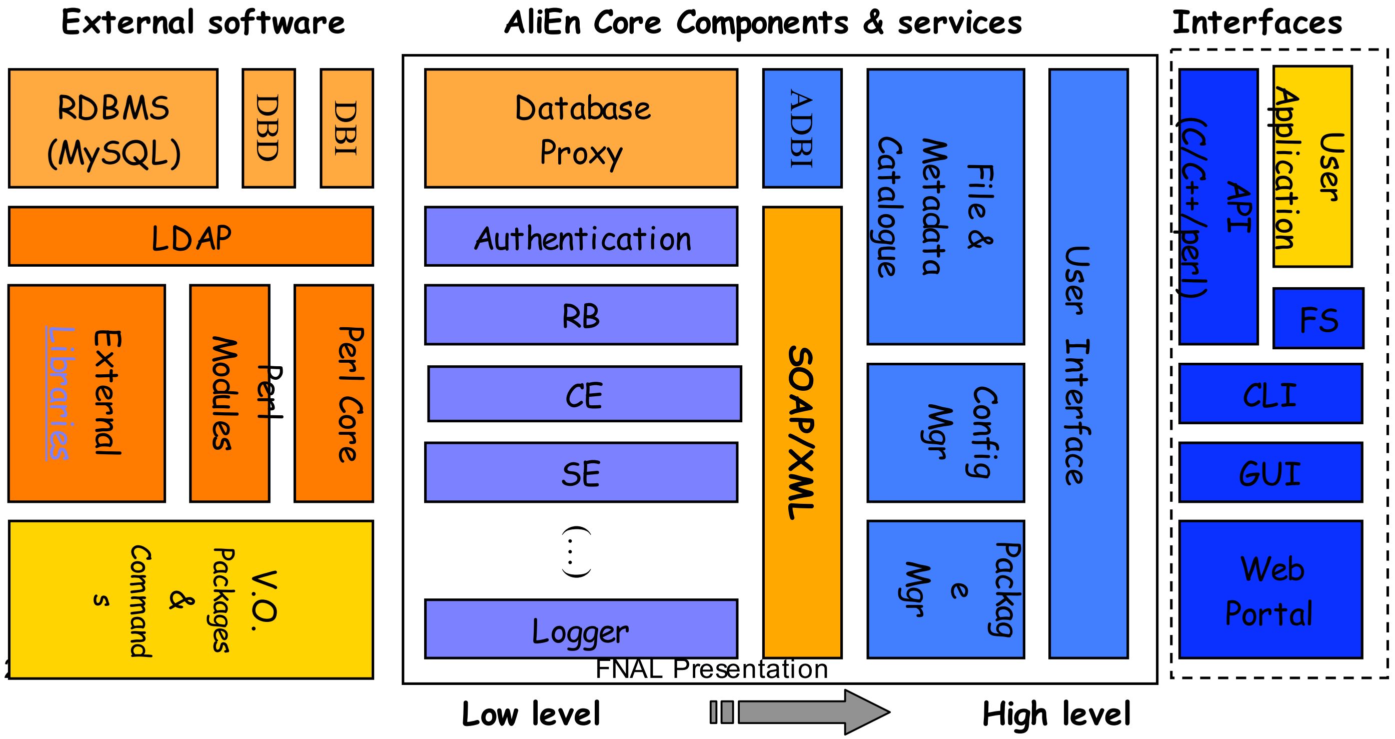
- The person who should give this talk is P.Buncic
  - Unfortunately he could not accompany us in this trip
- I am acting as a faithful proxy
  - Properly delegated credentials
  - Limited capabilities
- All the good ideas are his
- Whatever may be wrong in this slides is mine

The beginning

- In 2001 ALICE needed to do large productions
  - Completion of the detector TDR’s
  - Initial studies on the physics performance of the detector in preparation of the Physics Performance Report
- EDG could have been the natural choice for a mostly-EU based experiment
  - But it just started and it could not provide the necessary tools
- The boundary conditions were
  - Chronic understaffing of the computing project
  - Need for fairly large resources
- A perfect situation for a “Grid solution”

The ALICE Approach (AliEn)

- Millions lines of code in the OS domain dealing with Grid issues
- Why not using these to build the *minimal GRID* that *does the job*?
  - Fast development of a prototype, no problem in exploring new roads, restarting from scratch etc etc
  - Hundreds of users and developers for the modules
  - Immediate adoption of emerging standards
- AliEn (5% of code developed, 95% imported)



External software

RDBMS (MySQL)

DBD

DBI

LDAP

External Libraries

Perl Modules

Perl Core

V.O. Packages & Commands

AliEn Core Components & services

Database Proxy

Authentication

RB

CE

SE

(...)

Logger

ADBI

SOAP/XML

File & Metadata Catalogue

Config Mgr

Package Mgr

User Interface

Interfaces

API (C/C++/perl)

User Application

FS

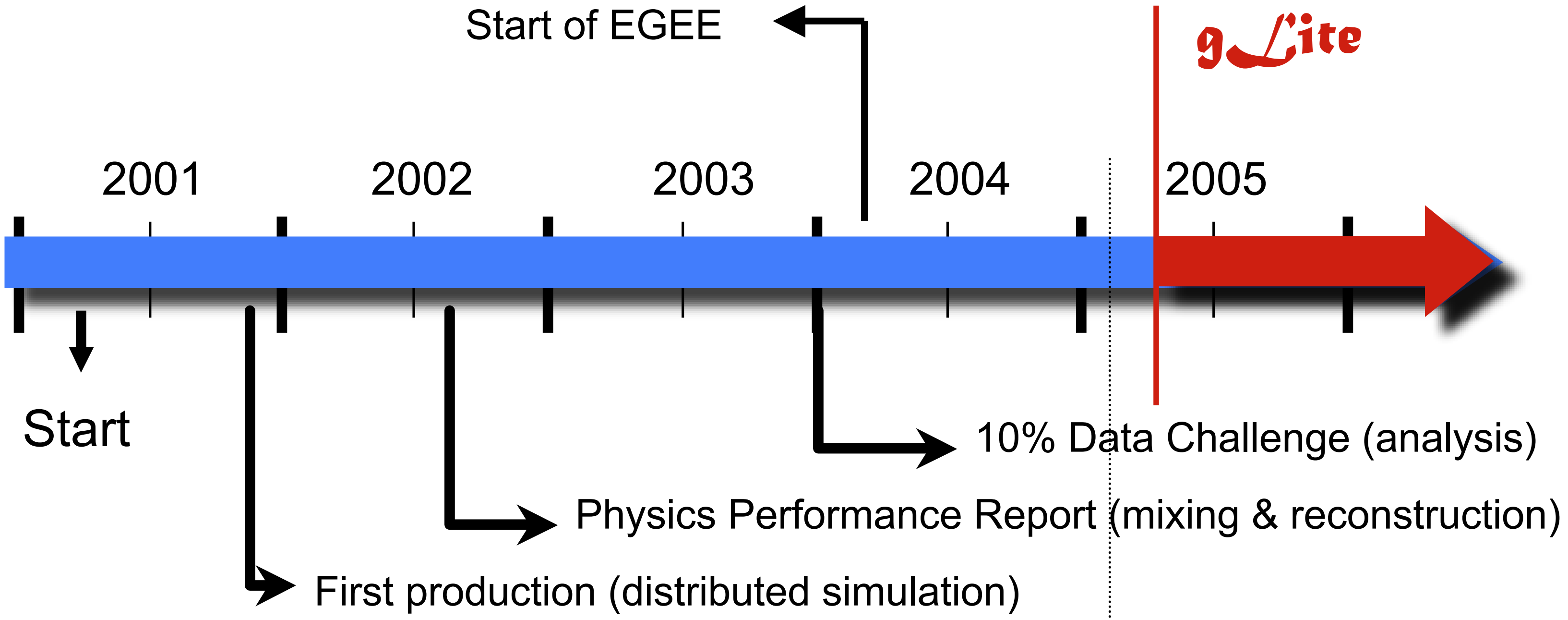
CLI

GUI

Web Portal

Low level → High level

The AliEn timeline



Start of EGEE

gLite

2001 2002 2003 2004 2005

Start

10% Data Challenge (analysis)

Physics Performance Report (mixing & reconstruction)

First production (distributed simulation)

Functionality + Simulation ⇨ Interoperability + Reconstruction ⇨ Performance, Scalability, Standards + Analysis

Experience with PDC 04

- Test and validate the ALICE computing model
  - Produce and analyse ~10% of the data of a standard data-taking year
  - Use the complete offline chain: AliEn, AliROOT, LCG and in Phase 3 – gLite+PROOF and the ALICE ARDA analysis prototype
  - ***Test*** of the software and ***physics analysis*** of the data for the PPR
- ***Do all of the above ENTIRELY on the GRID***
- Structure – divided in three phases:
  - Phase 1 - Production of underlying Pb+Pb and p+p events
    - Completed on time June 2004
  - Phase 2 - Mixing of different signal events with underlying Pb+Pb events (up to 50 times)
    - Completed on time September 2004
  - Phase 3 – Distributed analysis
    - Suspended

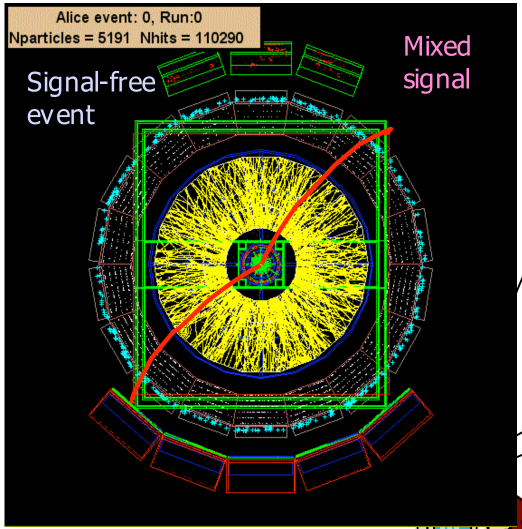
PDC04 schema

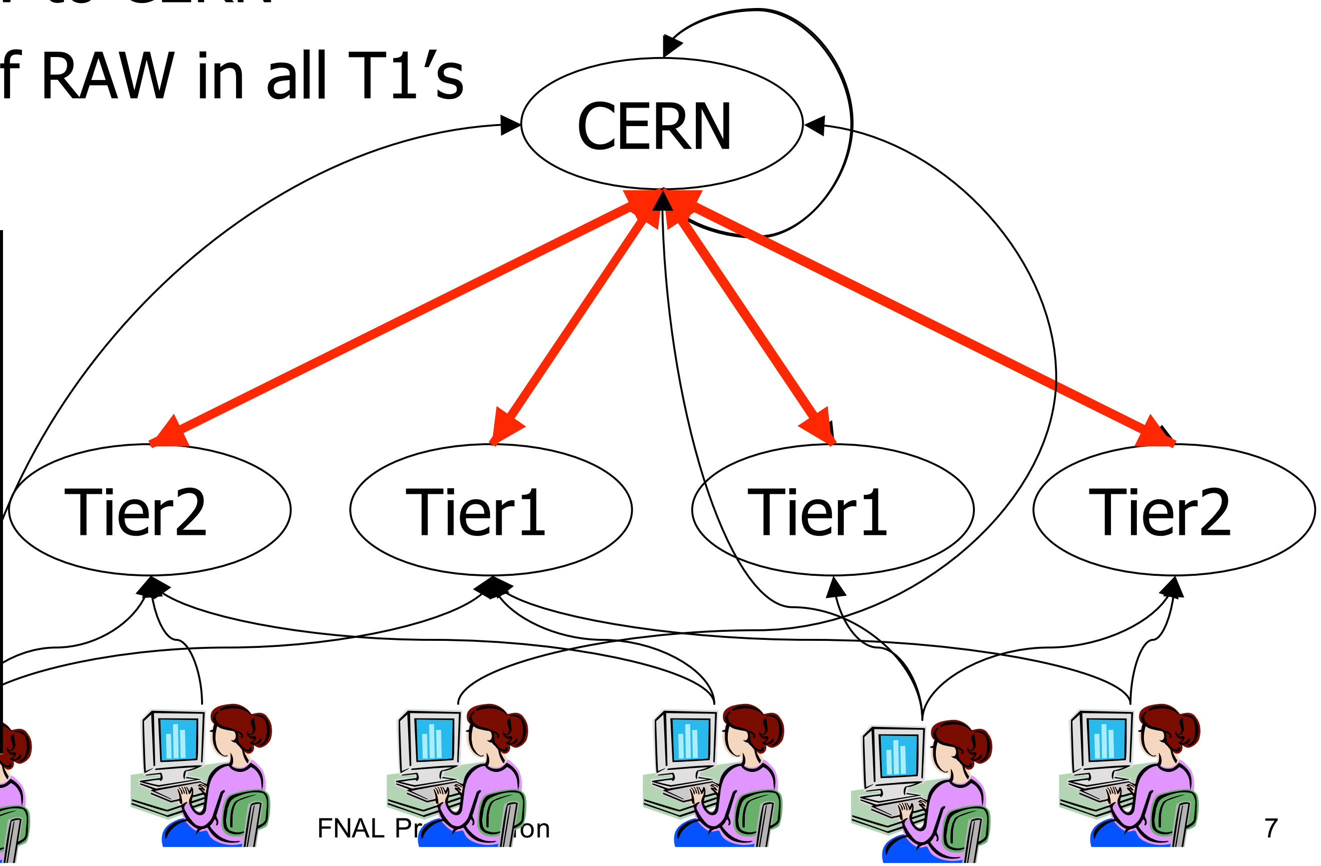
Production of RAW

Shipment of RAW to CERN

Reconstruction of RAW in all T1's

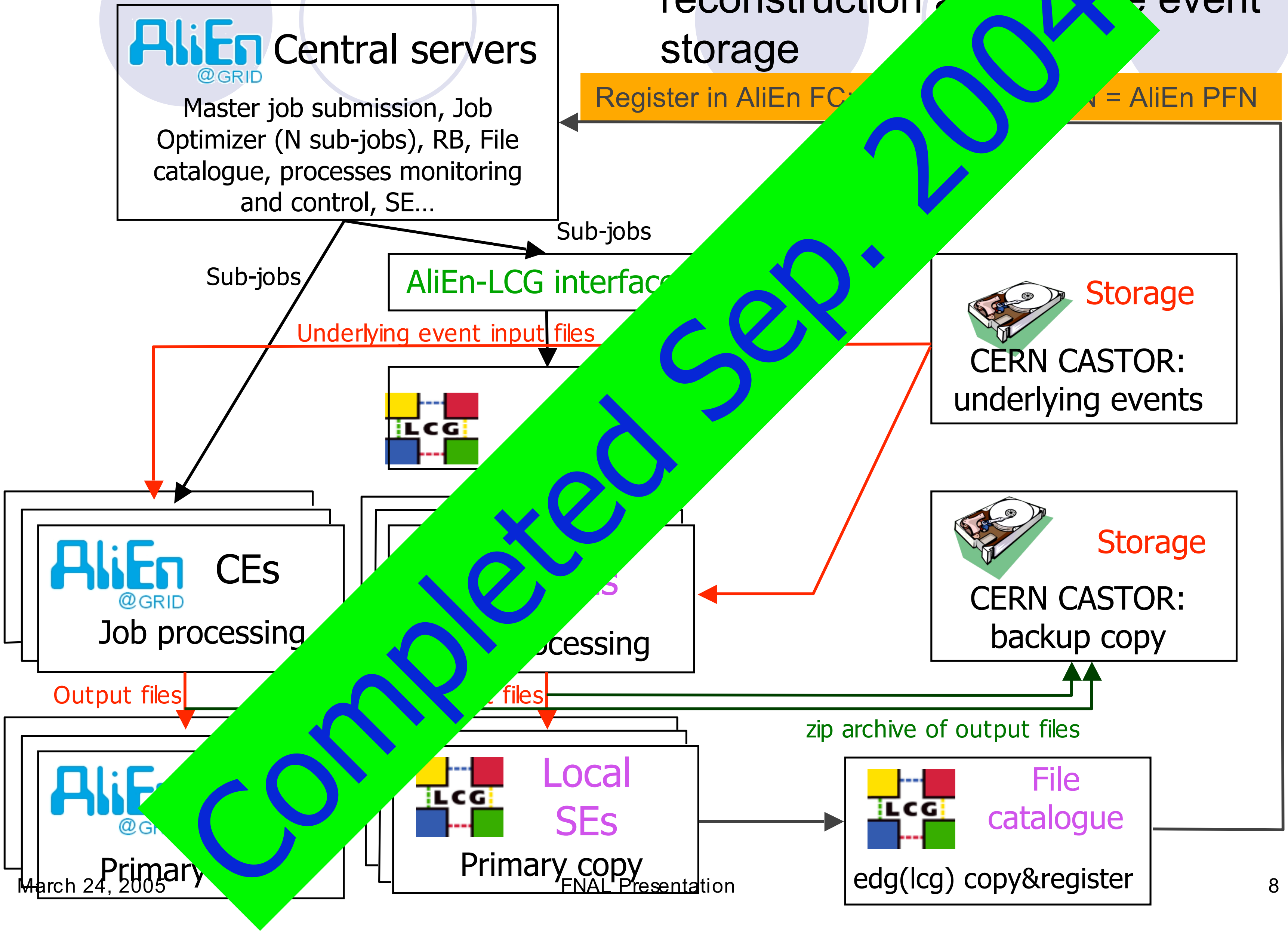
Analysis





Phase 2 job structure

- Task - simulate th ent reconstruction a e event storage



Central servers

Master job submission, Job Optimizer (N sub-jobs), RB, File catalogue, processes monitoring and control, SE…

Register in AliEn FC: N = AliEn PFN

Sub-jobs

Sub-jobs

AliEn-LCG interfac

Underlying event input files

Storage

CERN CASTOR: underlying events

CEs

Job processing

cessing

Storage

CERN CASTOR: backup copy

Output files

files

zip archive of output files

Local SEs

Primary copy

Primary

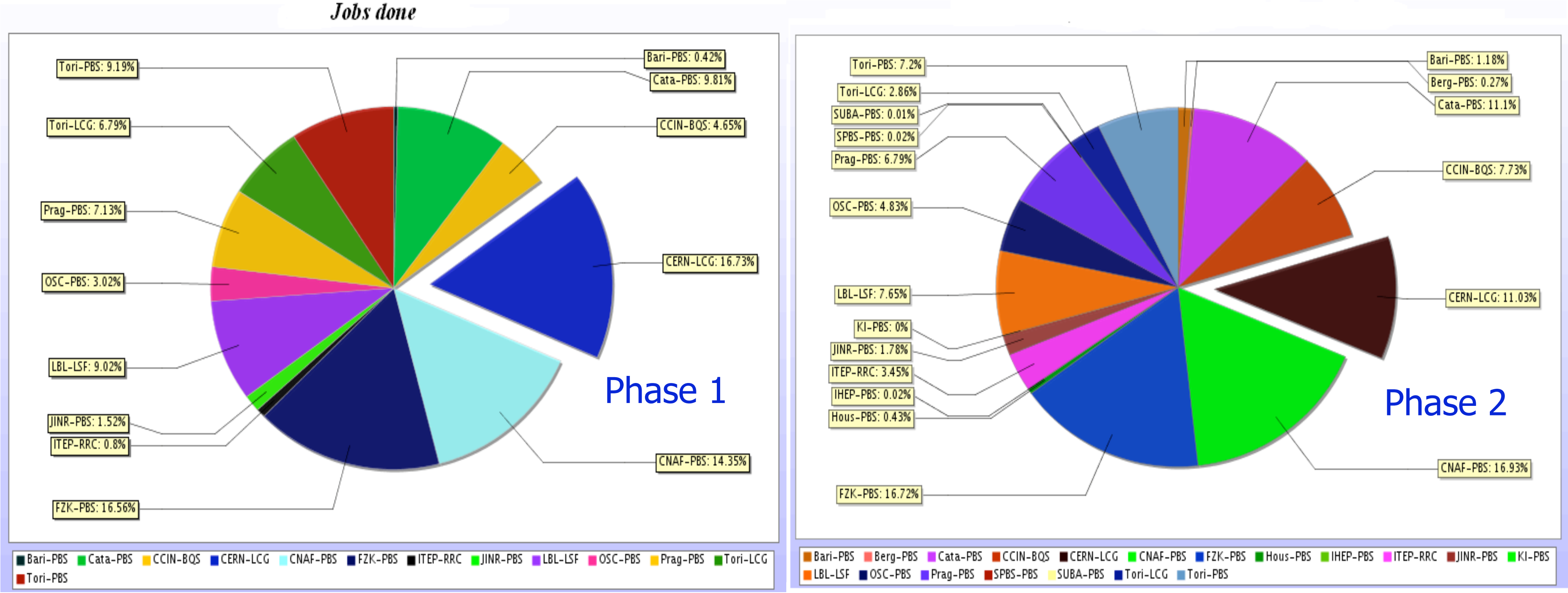
File catalogue

edg(lcg) copy&register

Completed Sep. 2004

Job repartition

- Jobs (AliEn/LCG): Phase 1 - 75/25%, Phase 2 – 89/11%
- More operation sites added to the ALICE GRID as PDC progressed



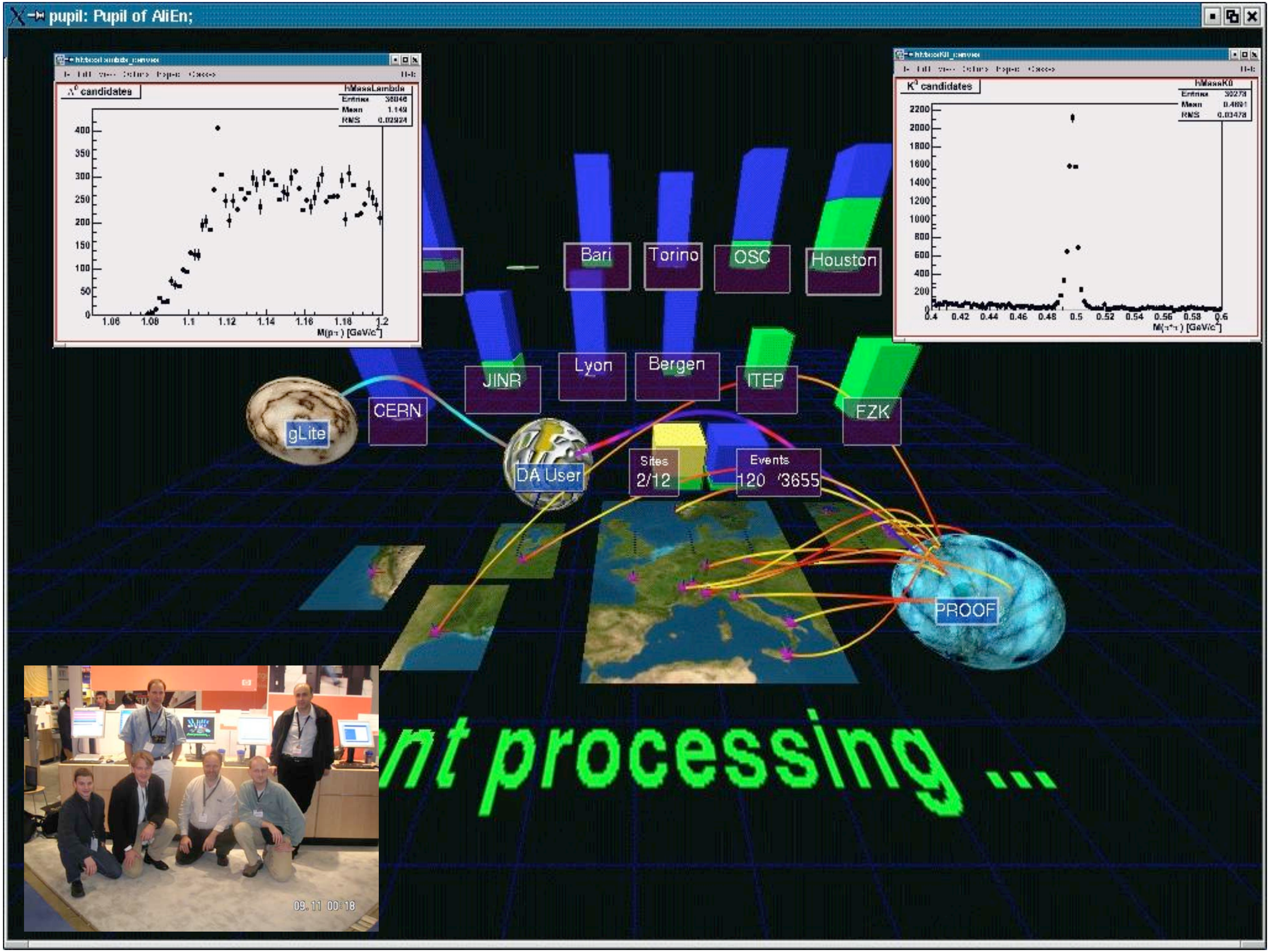
- 17 permanent sites (33 total) under AliEn direct control and additional resources through GRID federation (LCG)

Summary of PDC’04

- Computing resources
  - It took some effort to ‘tune’ the resources at the remote computing centres
  - Very positive response – CPU and storage increased during the PDC
- Middleware
  - AliEn proved fully capable of executing complex jobs on large amounts of resources
  - Functionality for Phase 3 has been demonstrated, but cannot be used
  - LCG MW adequate for Phase 1, but not for Phase 2 and in a competitive environment
  - It cannot provide the additional functionality needed for Phase 3
- Statistics
  - 400 000 jobs, 6 hours/job, 750 MSi2K hours
  - 9M entries in the AliEn file catalogue
  - 4M physical files at 20 AliEn SEs in centres world-wide
  - 30 TB at CERN CASTOR, 10 TB at remote AliEn SEs & backup at CERN
  - 200 TB network transfer CERN –> remote computing centres
  - AliEn efficiency observed >90%, LCG observed efficiency 60% (see GAG document)

Development of Analysis

- Analysis Object Data designed for efficiency
  - Contain only data needed for a particular analysis
- Analysis à la PAW
  - ROOT + at most a small library
- Work on the distributed infrastructure has been done by the ARDA project
- Batch analysis infrastructure
  - Prototype published at the end of 2004 with AliEn
- Interactive analysis infrastructure
  - Demonstration performed at the end 2004 with AliEn⇒gLite
- Physics working groups are just starting now, so timing is right to receive requirements and feedback

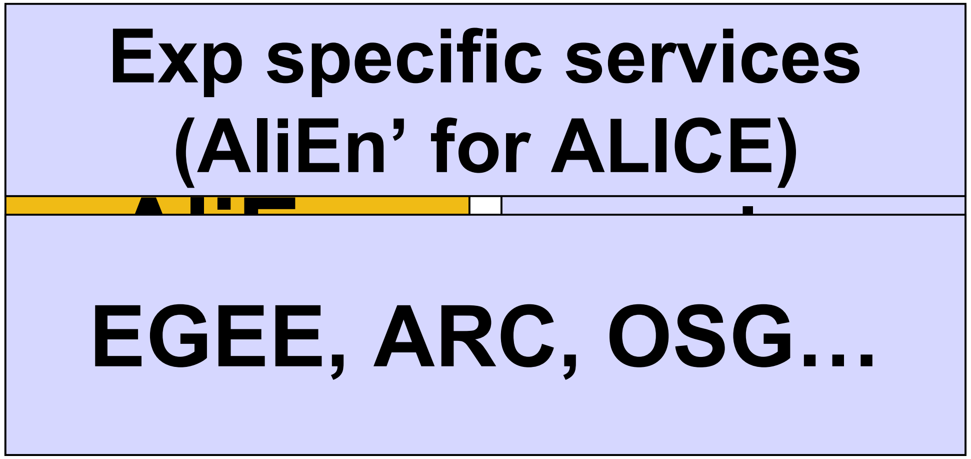


pupil: Pupil of AliEn;
Λ⁰ candidates
hMassLambda
Entries 36046
Mean 1.149
RMS 0.02924
400
350
300
250
200
150
100
50
0
1.06
1.08
1.1
1.12
1.14
1.16
1.18
1.2
M(pπ) [GeV/c²]
K⁰ candidates
hMassK0
Entries 30278
Mean 0.4891
RMS 0.03478
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0.4
0.42
0.44
0.46
0.48
0.5
0.52
0.54
0.56
0.58
0.6
M(π⁺π⁻) [GeV/c²]
Bari
Torino
OSC
Houston
JINR
Lyon
Bergen
ITEP
FZK
CERN
gLite
DA User
Sites
2/12
Events
120 /3655
PROOF
nt processing ...
09. 11 00:18

Short history of EGEE MW

- History
  - Oct'03: ARDA proposes to abandon EDG-derived MW and to take a new fresh start with an AliEn architecture
  - Mar '04: AliEn developers are hired by EGEE and start working on new MW
  - May '04: An AliEn-derived prototype (gLite) is offered to pilot users (ARDA, Biomed..)
  - Dec '04: Experiments ask for this prototype to be deployed on larger preproduction service as part of the EGEE release
  - Jan '05: Management decides that the AliEn-derived elements will not be in the release
- Current situation
  - EGEE intends to provide the same functionality of the AliEn-derived MW
    - But this implies a delay in the release schedule
    - The new components will have to be field tested
    - Most of the architecture stays the same -- AliEn based

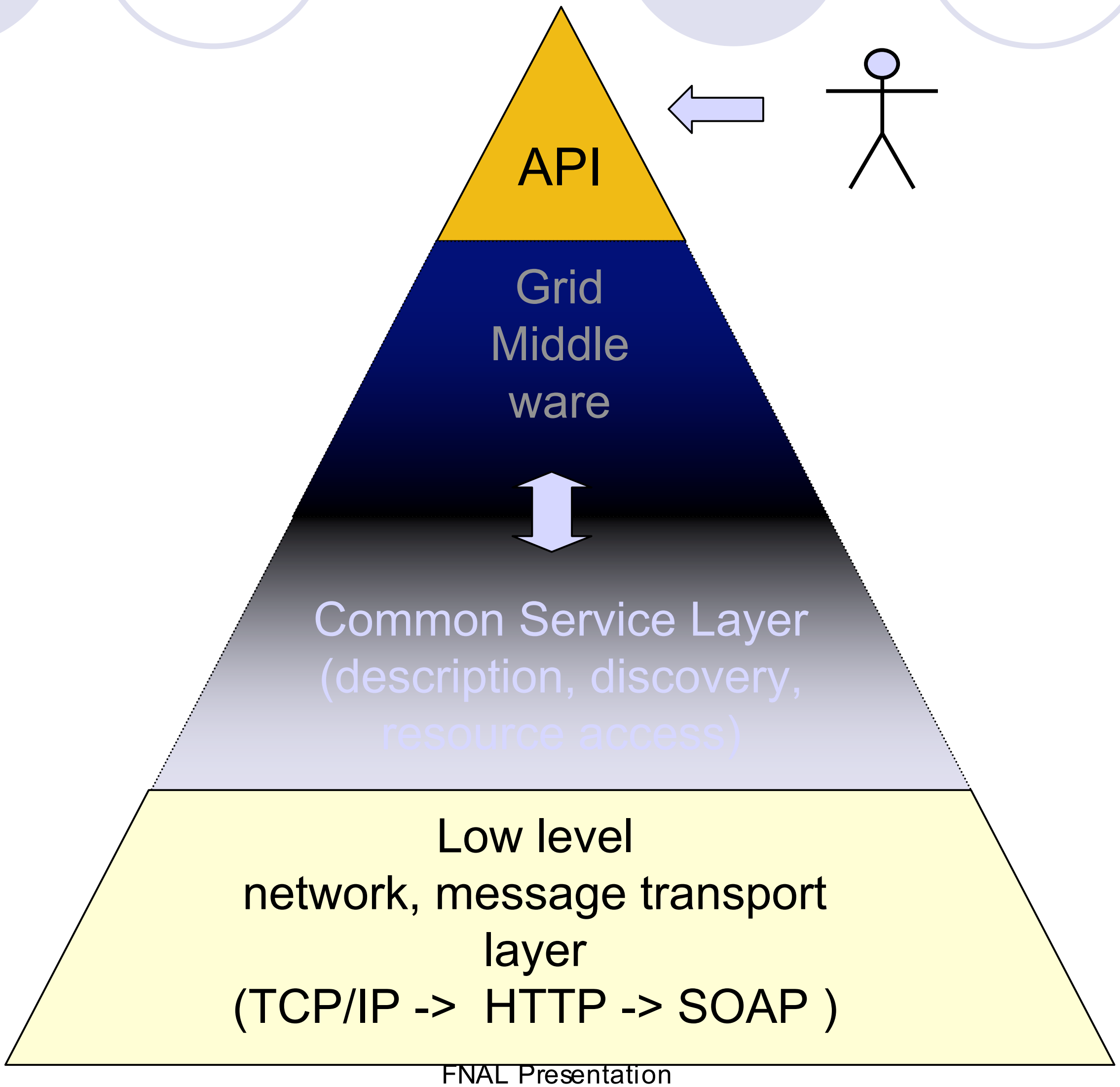
ALICE view on the current situation



Exp specific services (AliEn' for ALICE)

EGEE, ARC, OSG…

Grid Middleware



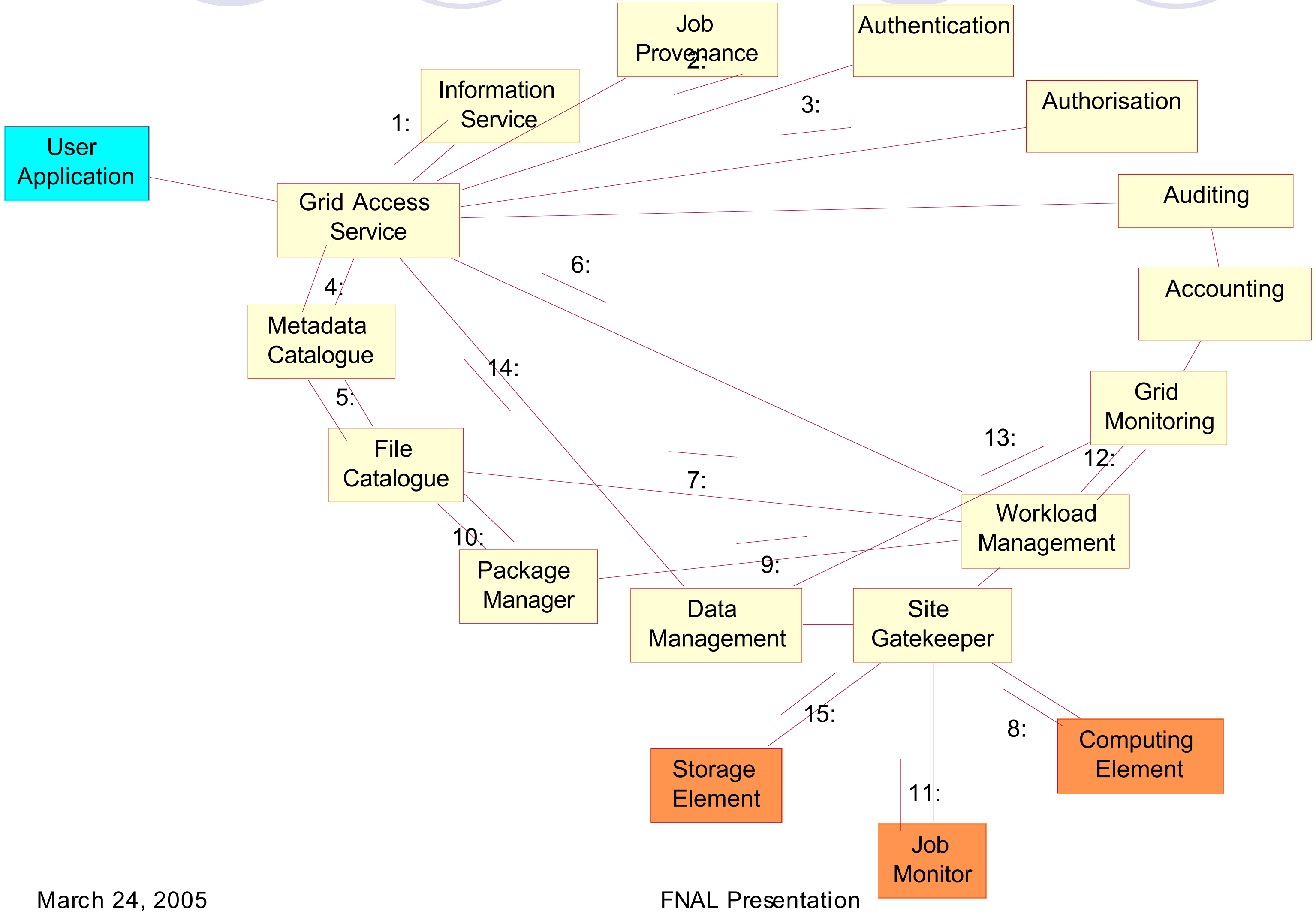
API

Grid Middle ware

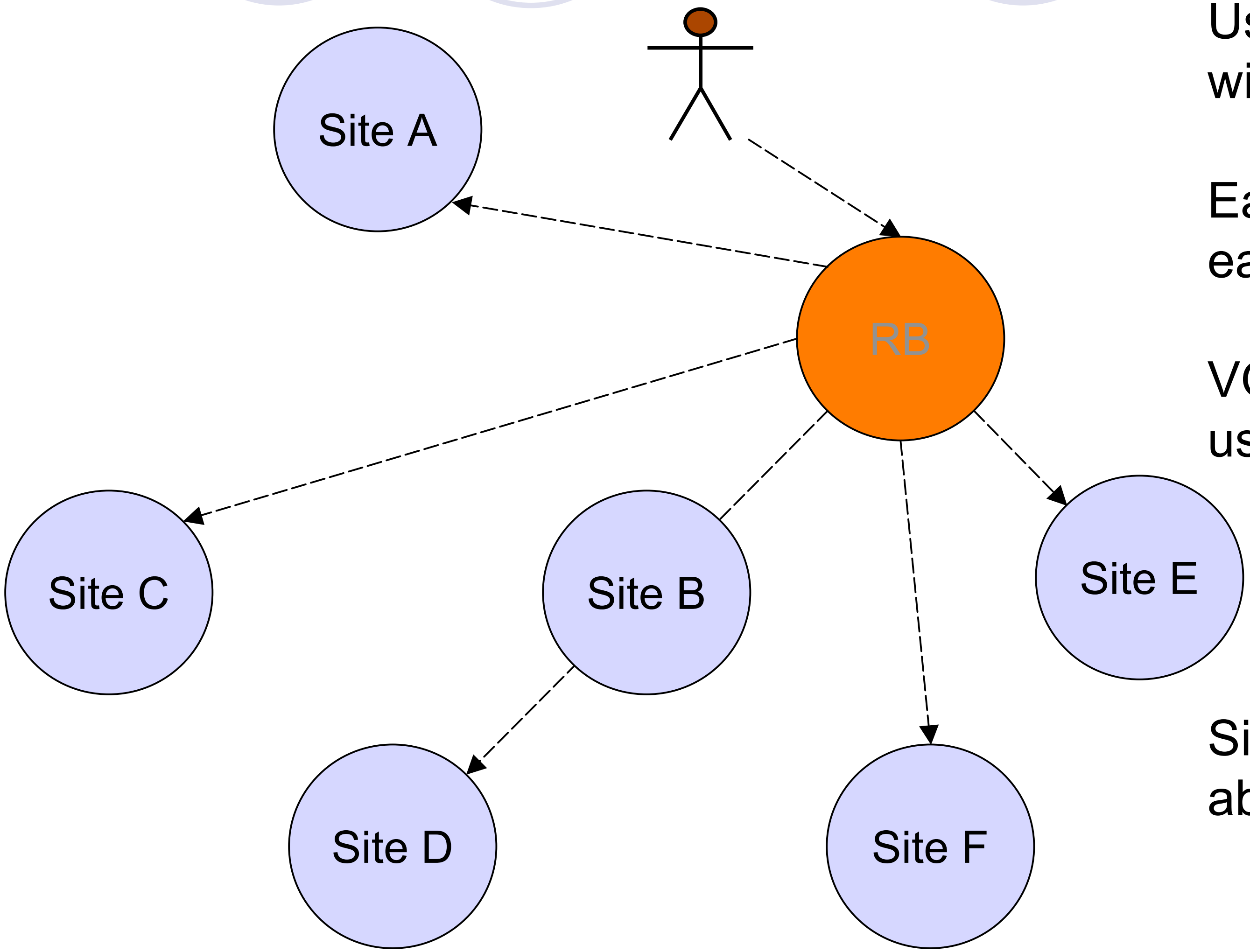
Common Service Layer (description, discovery, resource access)

Low level network, message transport layer (TCP/IP -> HTTP -> SOAP)

ARDA Service decomposition



Globus Model



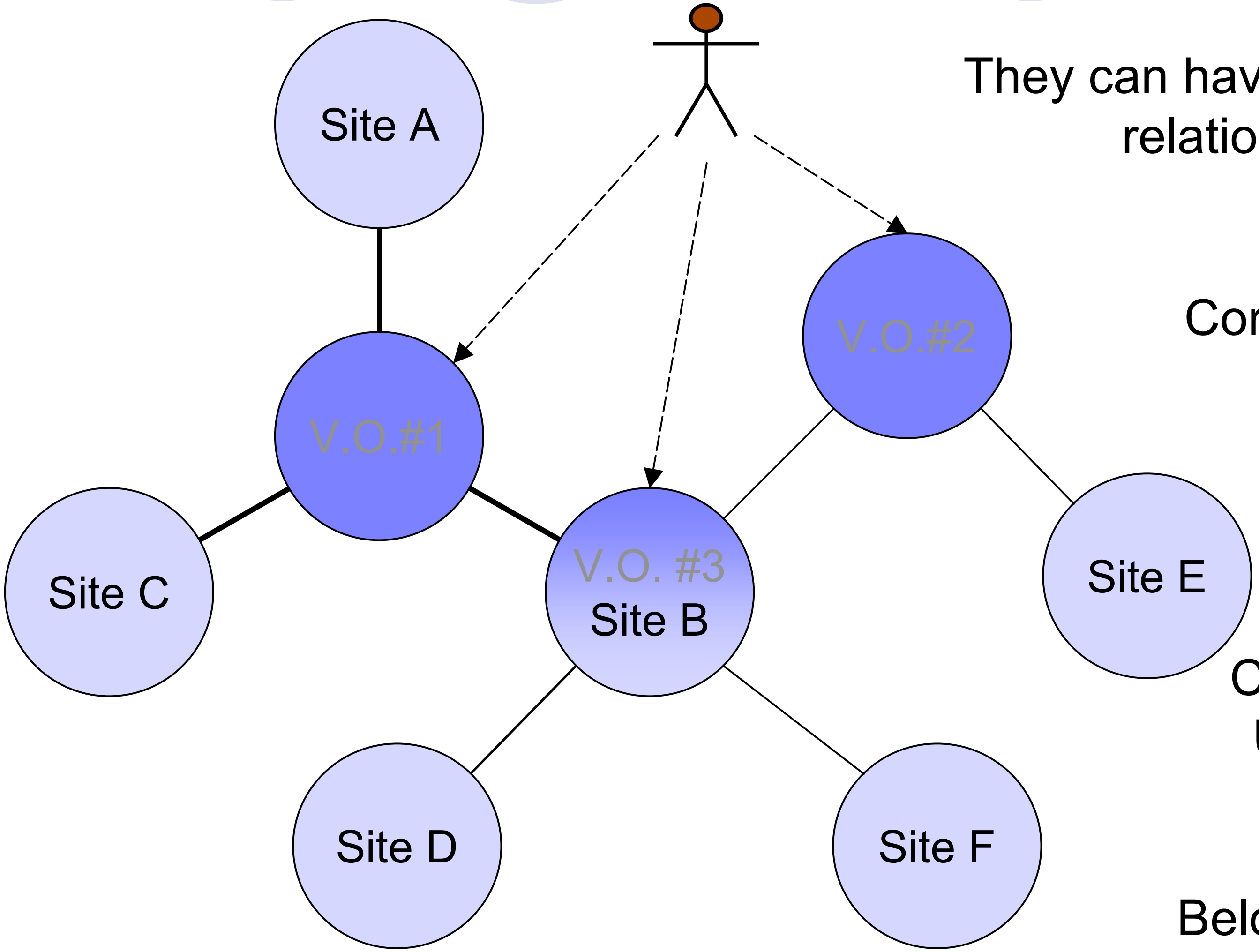
User interacts directly with site

Each site has to map each user to local id

VO is a group of users

Sites do not know about VOs (RB does)

Site, V.O. & GSP



VO can be created and deleted dynamically

They can have hierarchical relationships

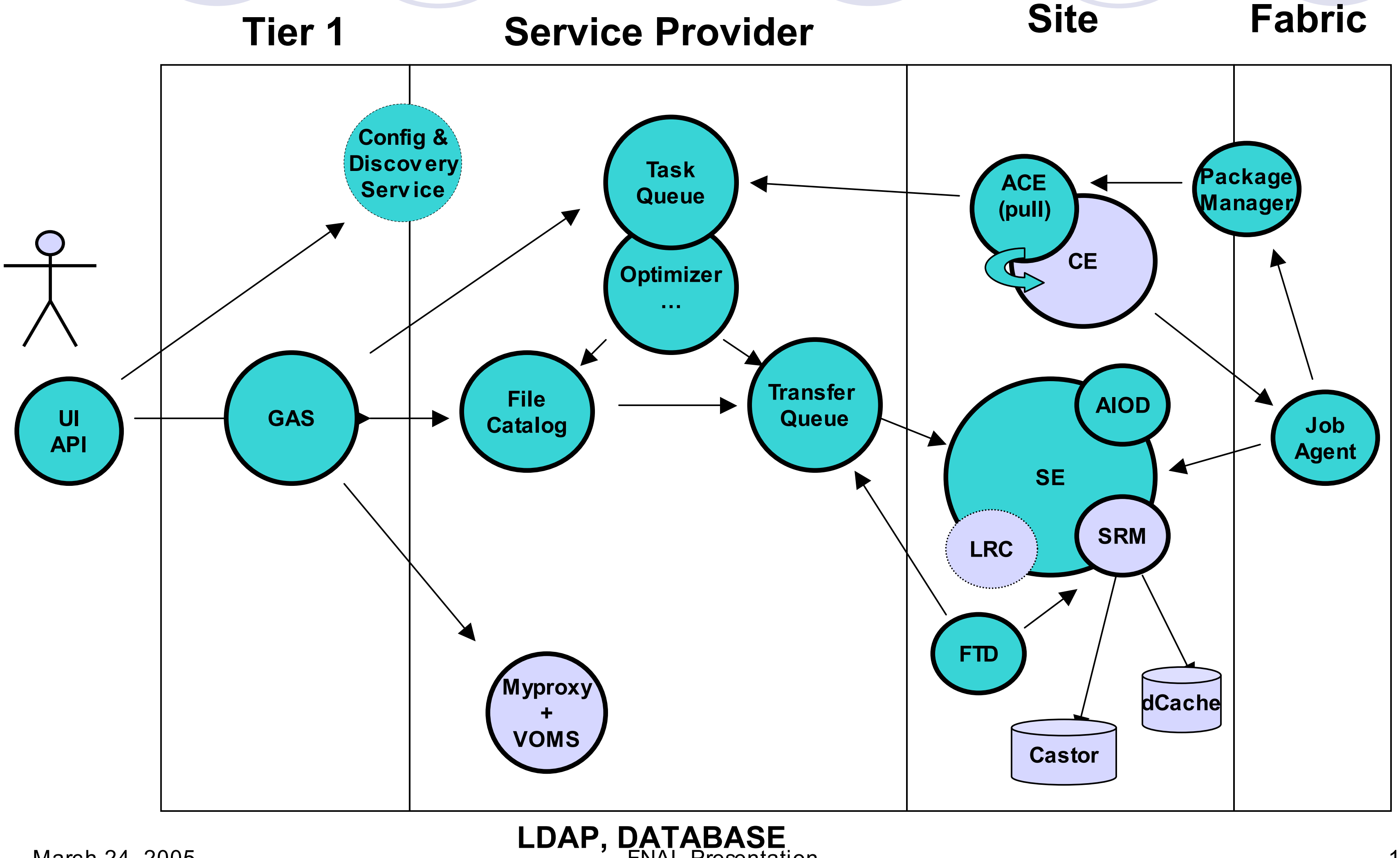
GSP:
Core Services (per VO)

Site:
CE, SE (per VO)

VO:
Collection of Sites, Users & Services

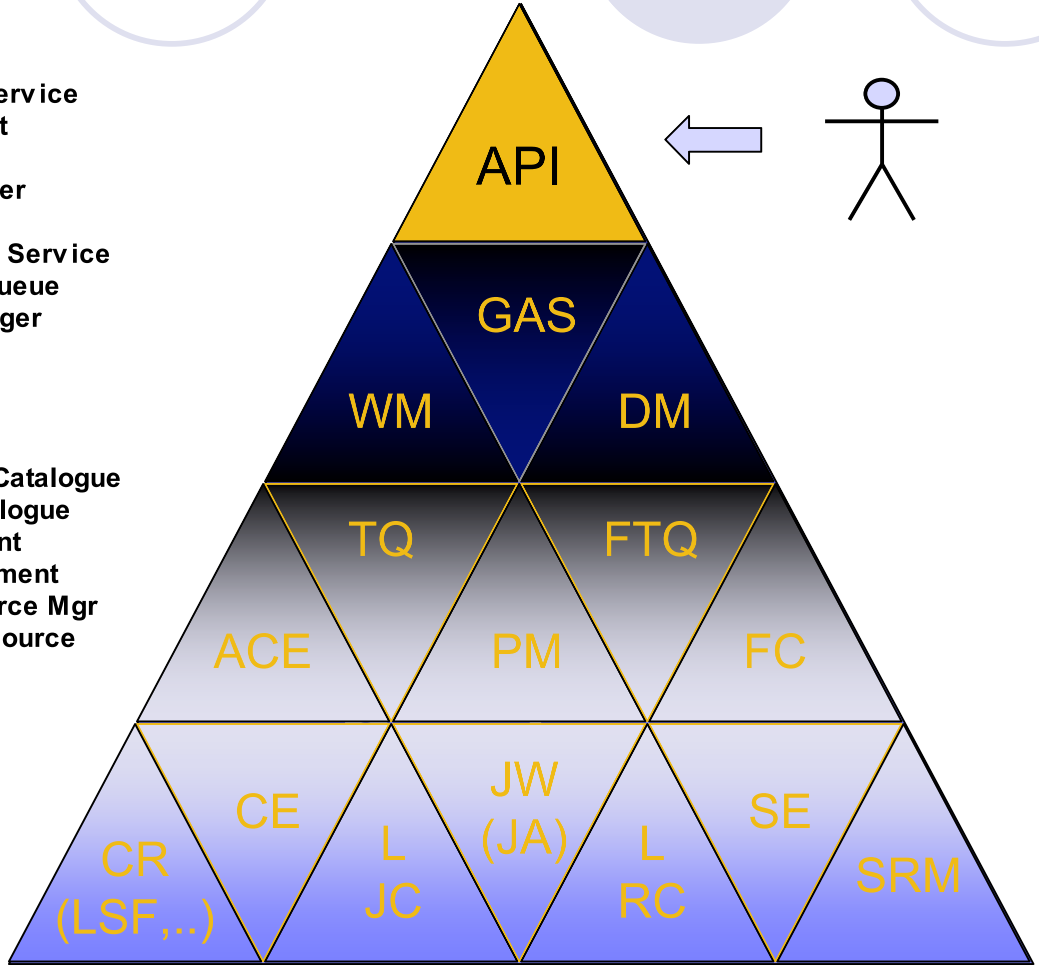
User:
Belongs to one or more VO’s

gLite Prototype (seen by ARDA)

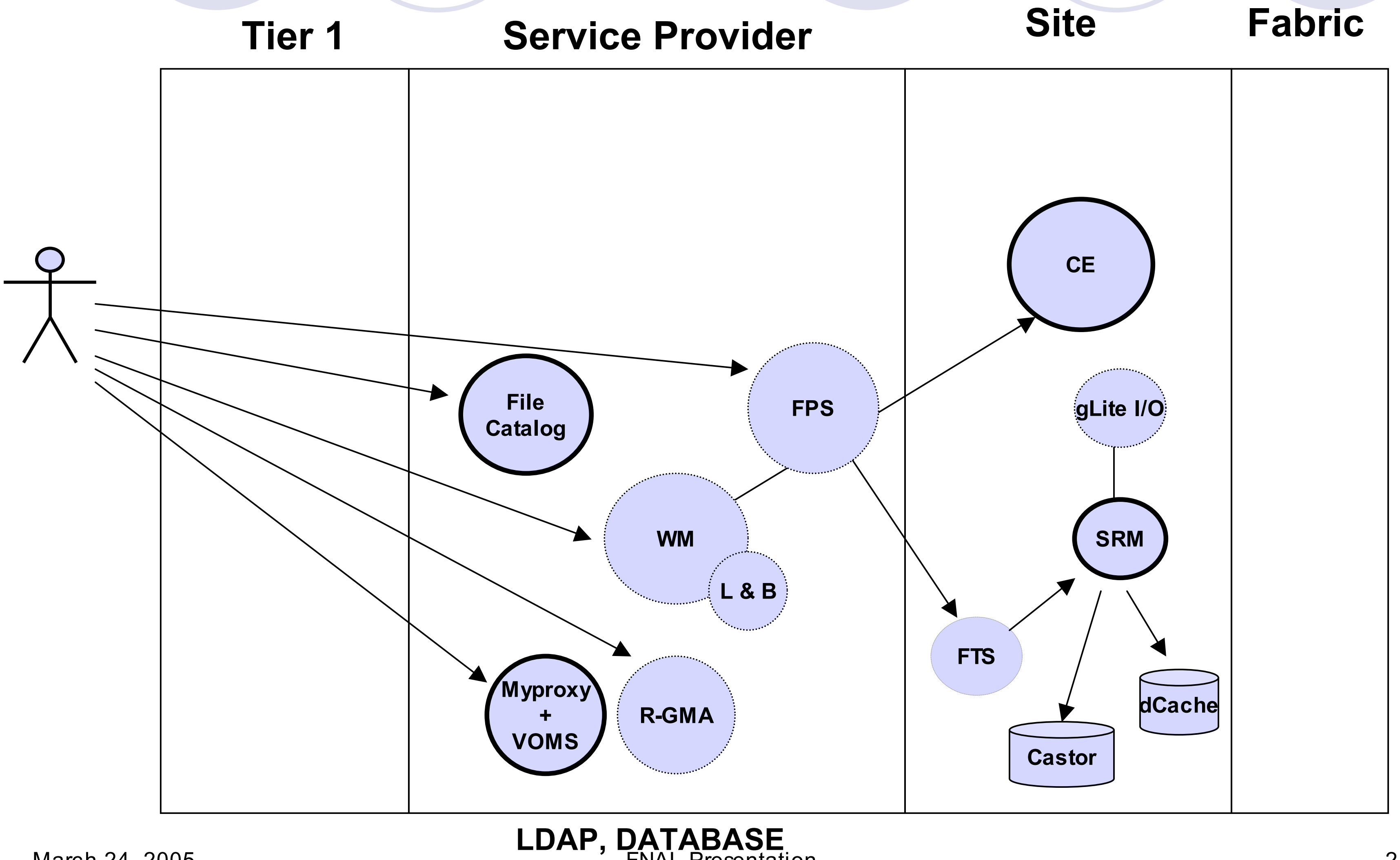


Middleware Services in AliEn

GAS	Grid Access Service
WM	Workload Mgmt
DM	Data Mgmt
RB	Resource Broker
TQ	Task Queue
FPS	File Placement Service
FQ	File Transfer Queue
PM	Package Manager
ACE	AliEn CE (pull)
FC	File Catalogue
JW	Job Wrapper
JA	Job Agent
LRC	Local Replica Catalogue
?	Local Job Catalogue
SE	Storage Element
CE	Computing Element
SRM	Storage Resource Mgr
CR	Computing Resource (LSF, PBS,…)

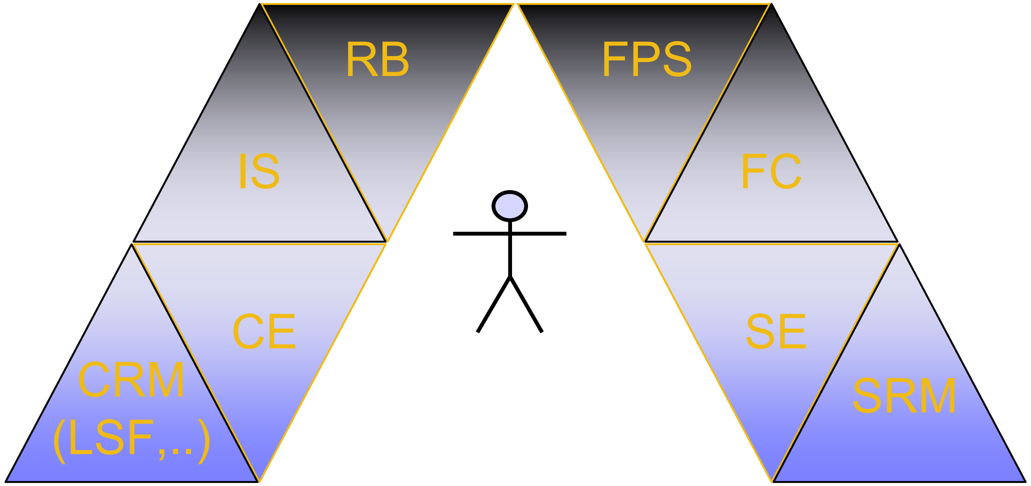


What will be delivered in gLite 1.0

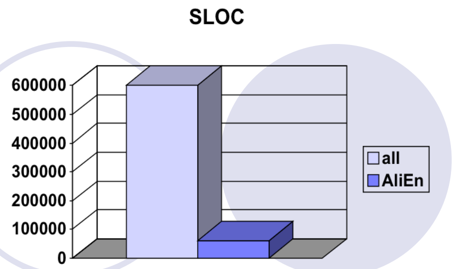


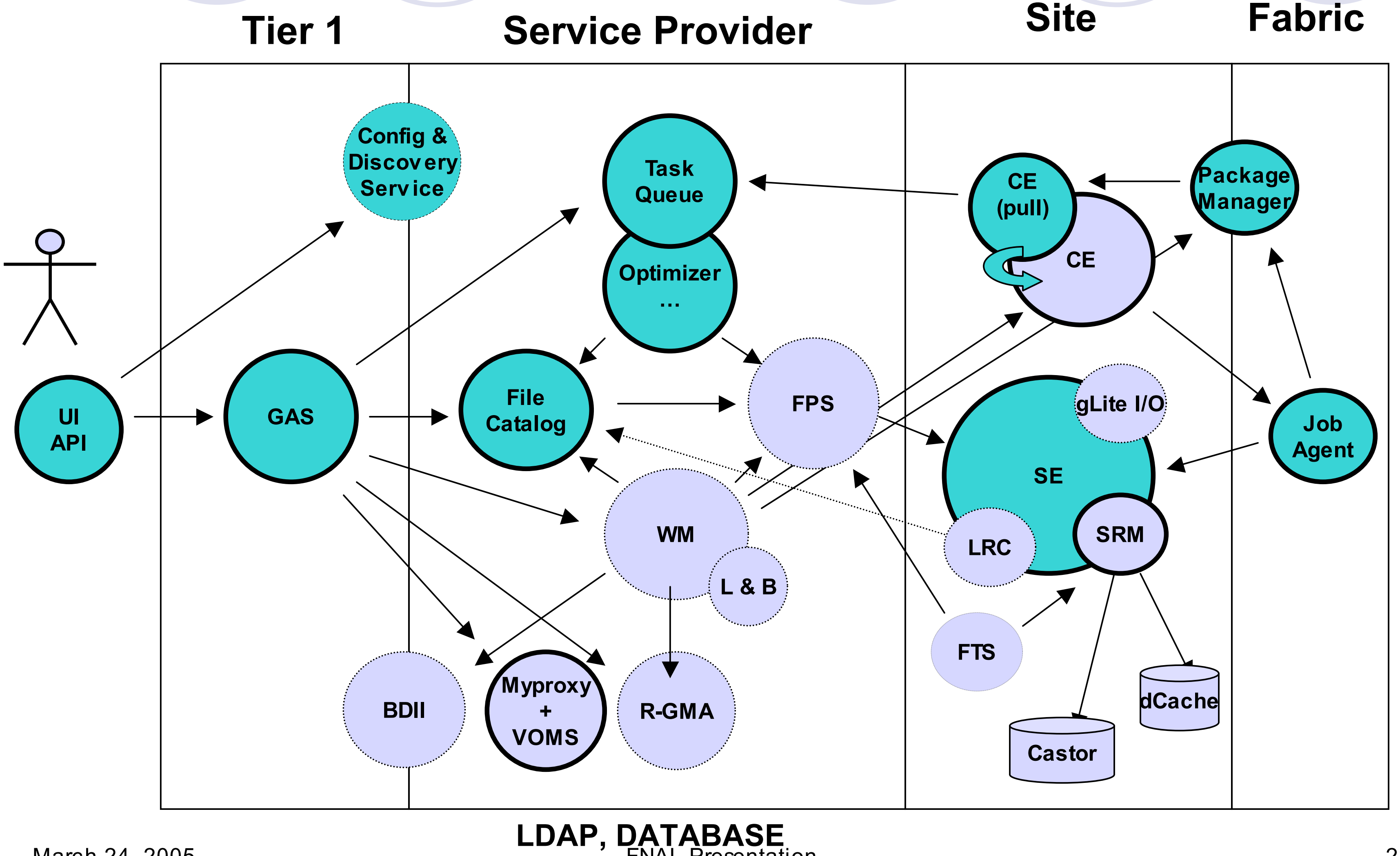
Middleware Services in gLite 1.0

The user has to interact directly with the services
Higher level services will have to be developed by the experiments to “fill the gaps”

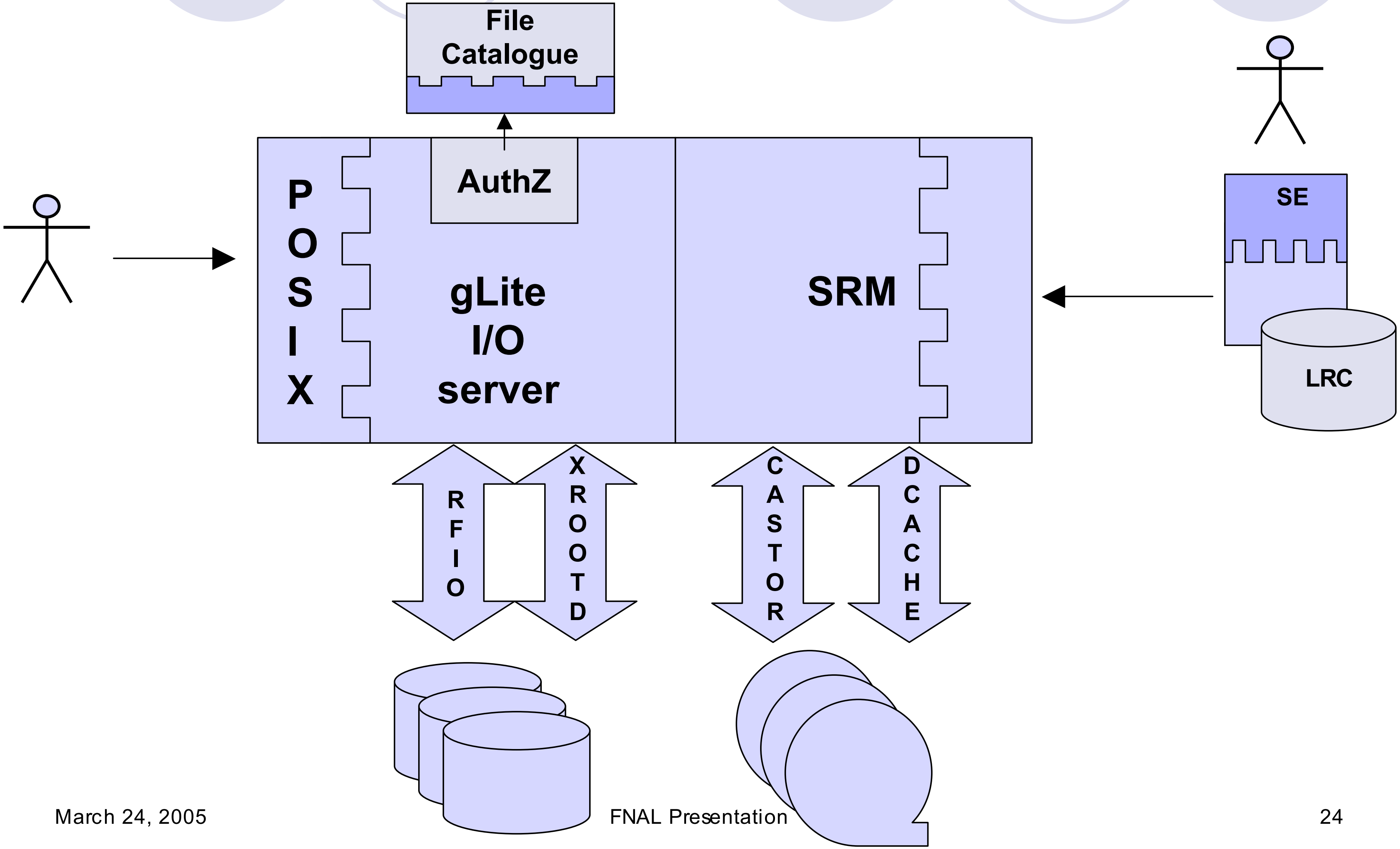


Abandoned gLite RC1

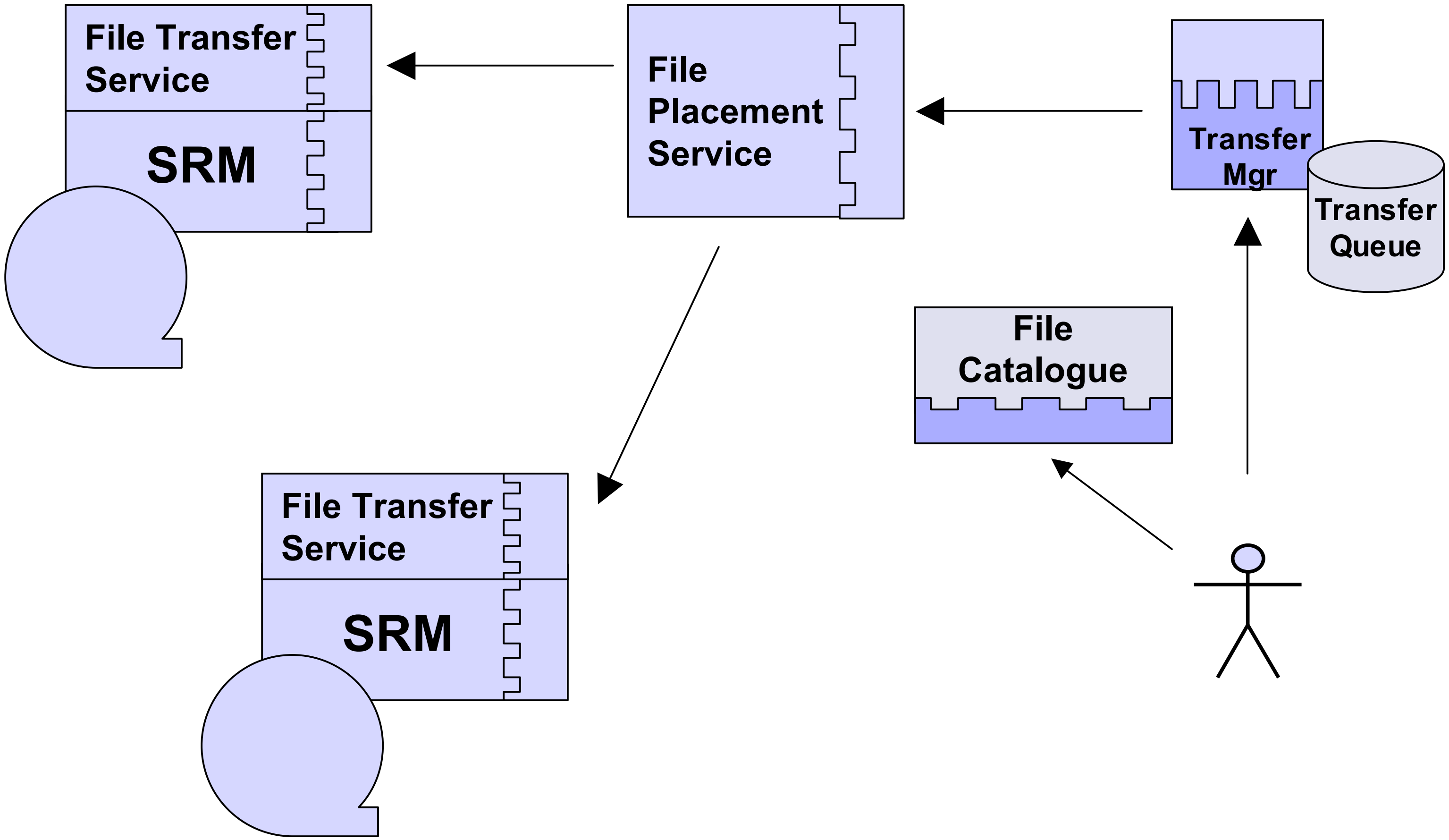




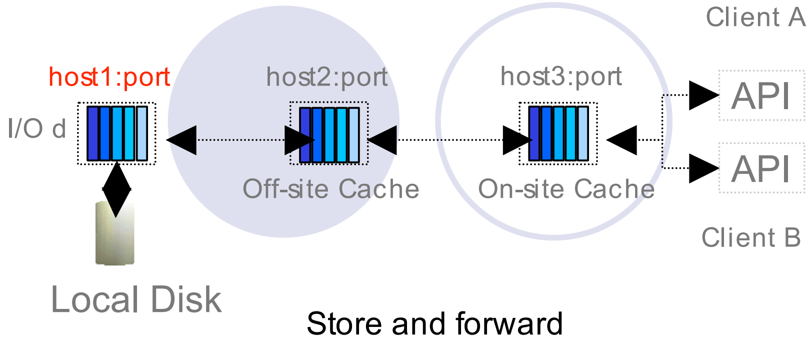
Site: Data Mgmt



VO Data Management

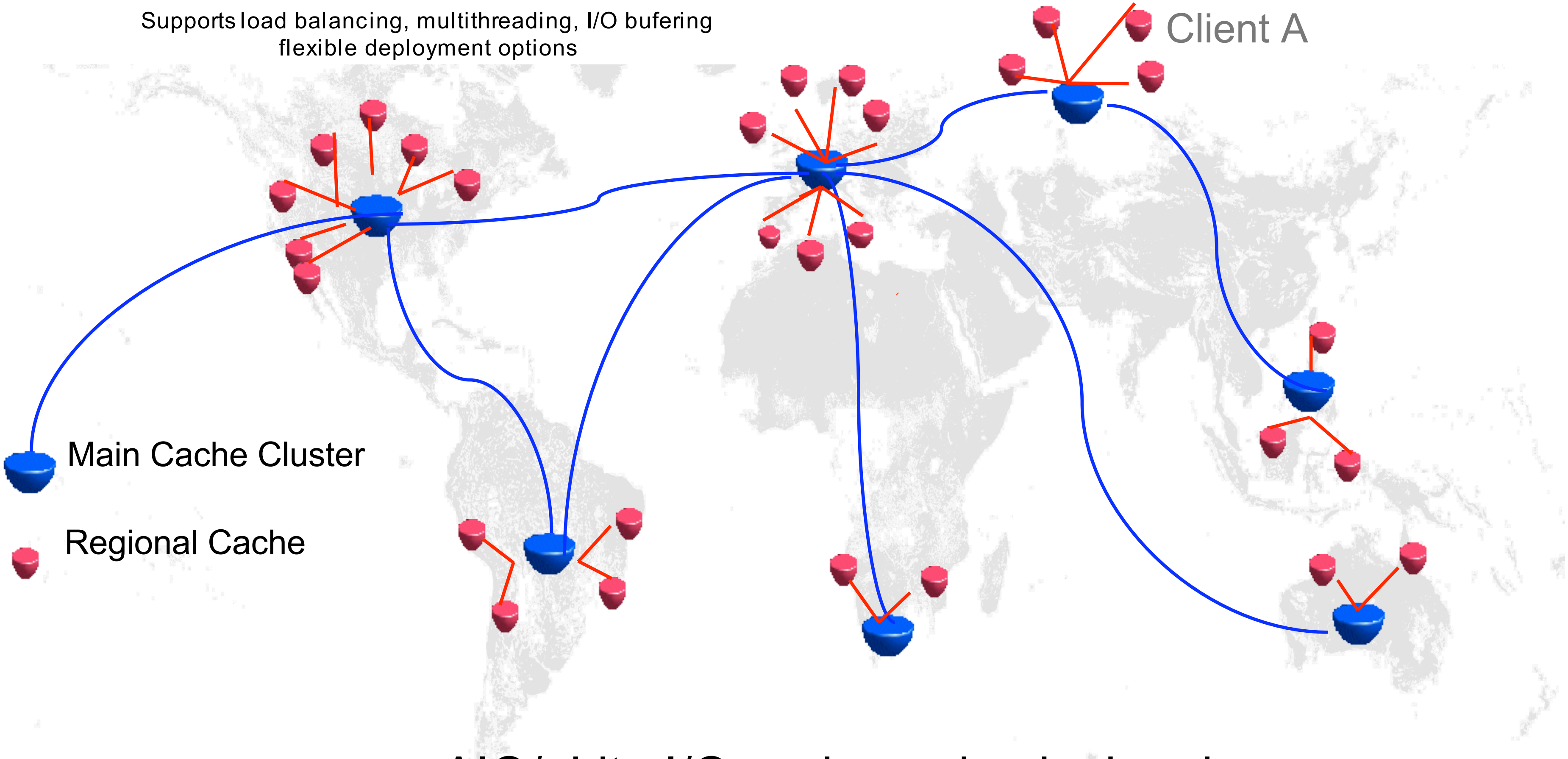


Crosslink-Cache gLite I/O



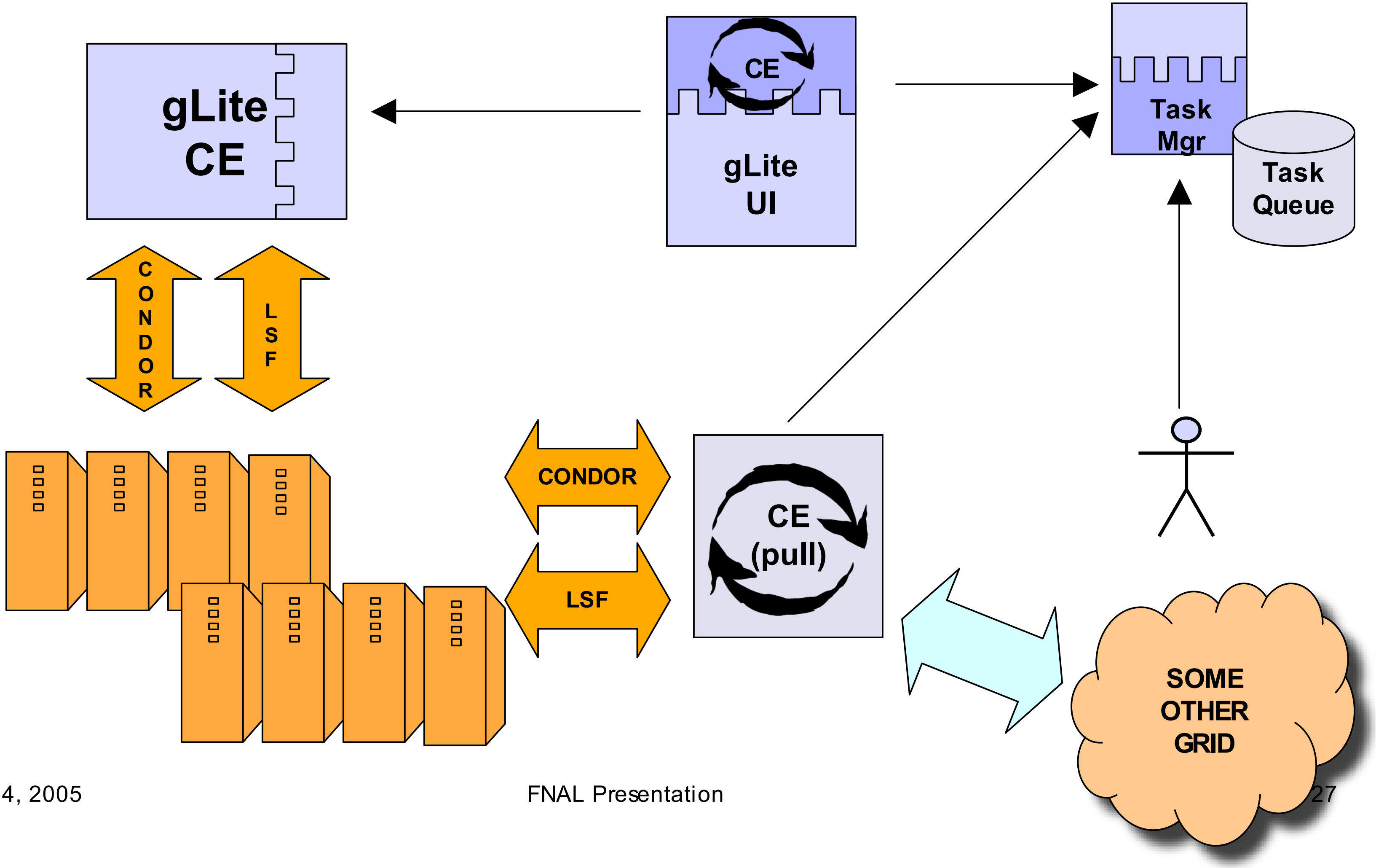
Store and forward

Supports load balancing, multithreading, I/O bufering flexible deployment options

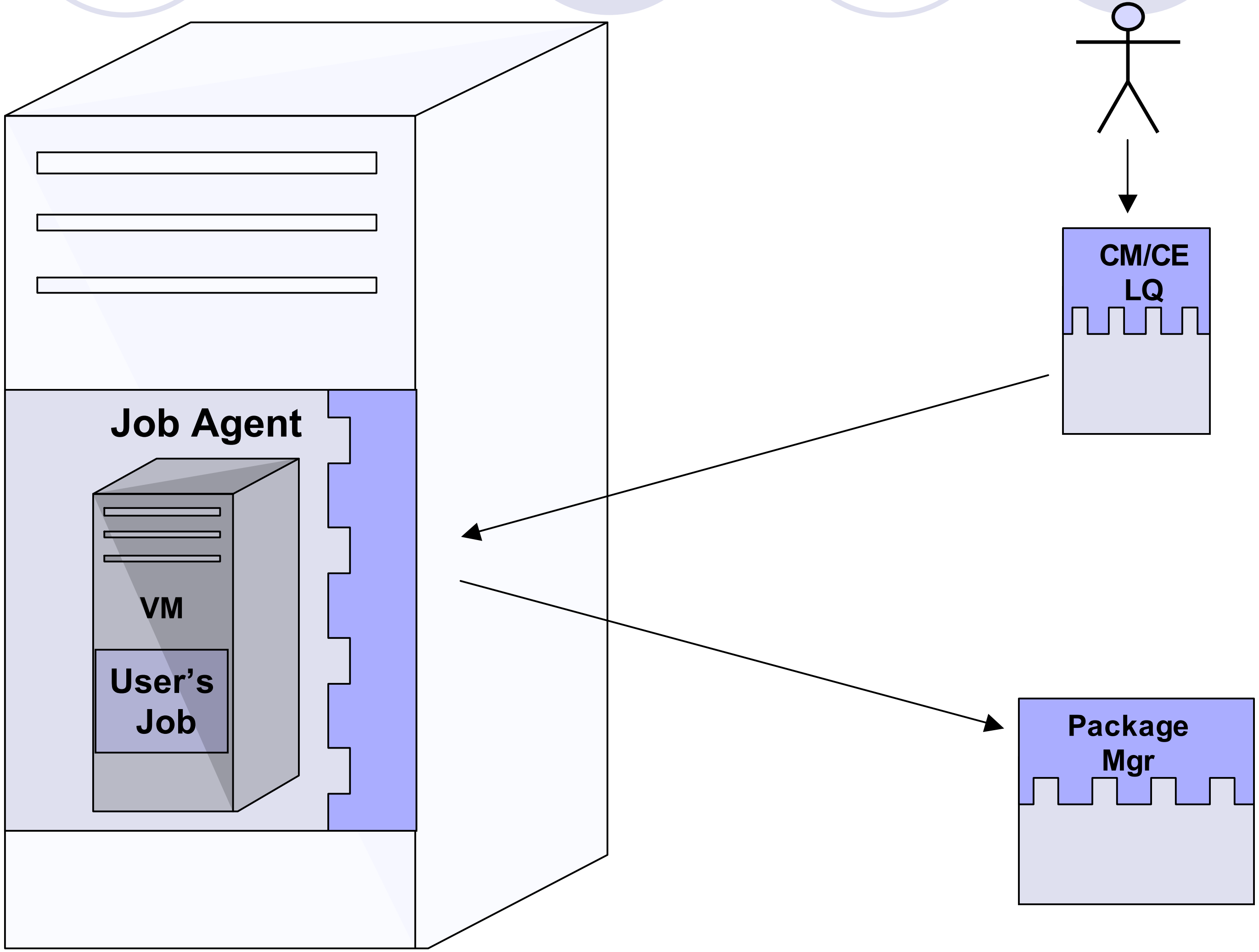


AIO/gLite I/O cache = shock absorber

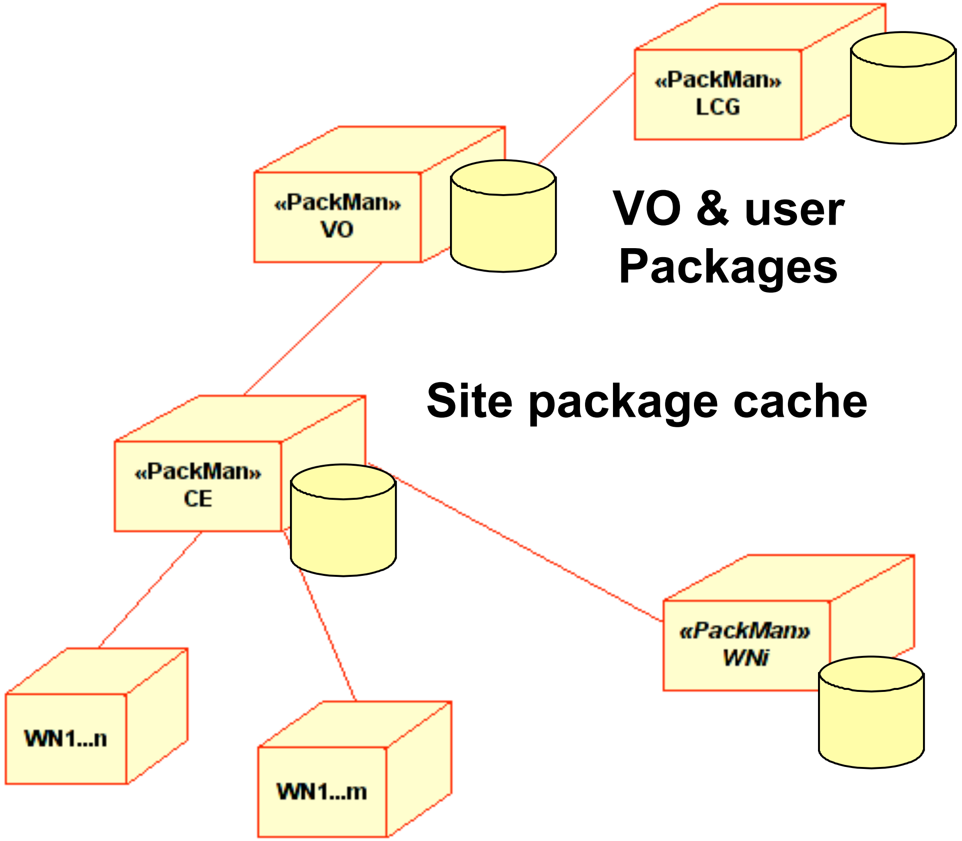
VO Job Management



Site: A possible evolution of the worker node



Package Manager Deployment



Common Packages (ROOT, POOL,..)

VO & user Packages

Site package cache

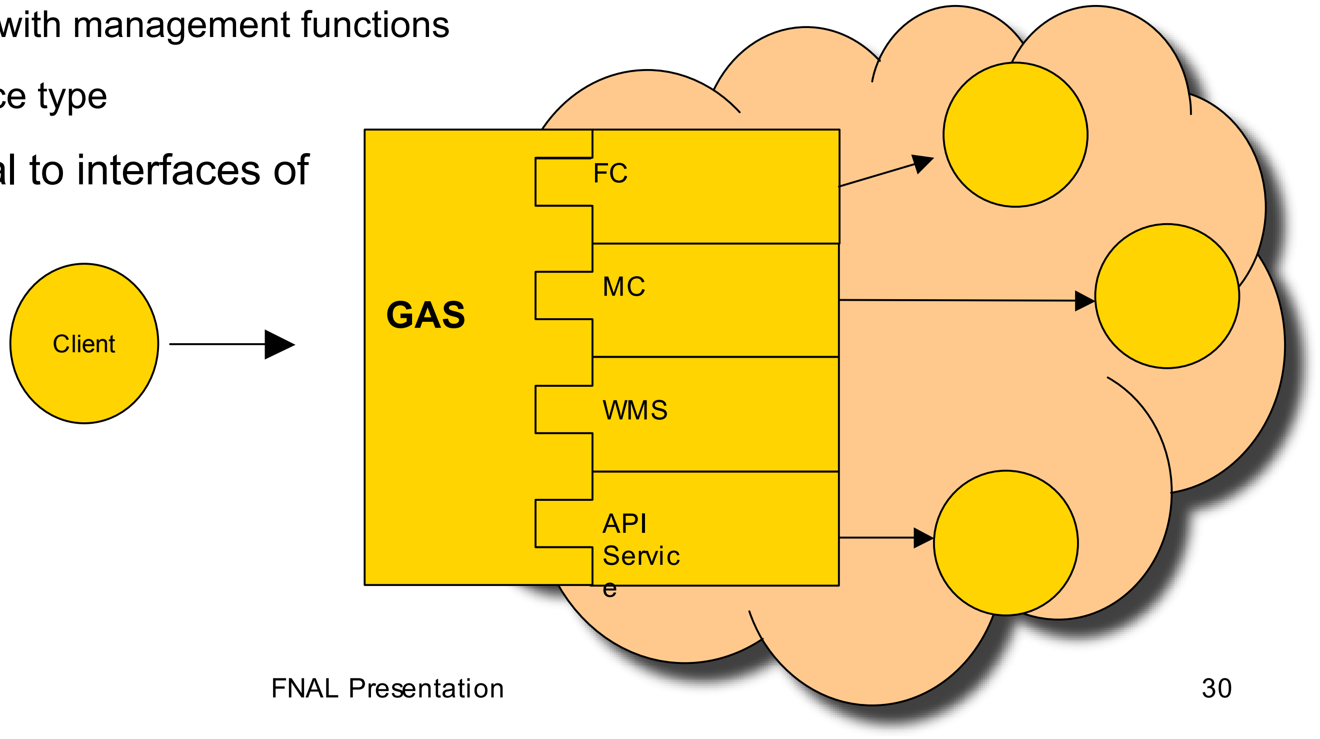
Worker node cache

GAS

- The Grid Access Service represents the user entry point to a set of core services

Components:
- General GAS Interface with management functions (destroy, renew, …)
- Snippets for each service type (FC, MC, WMS, …)

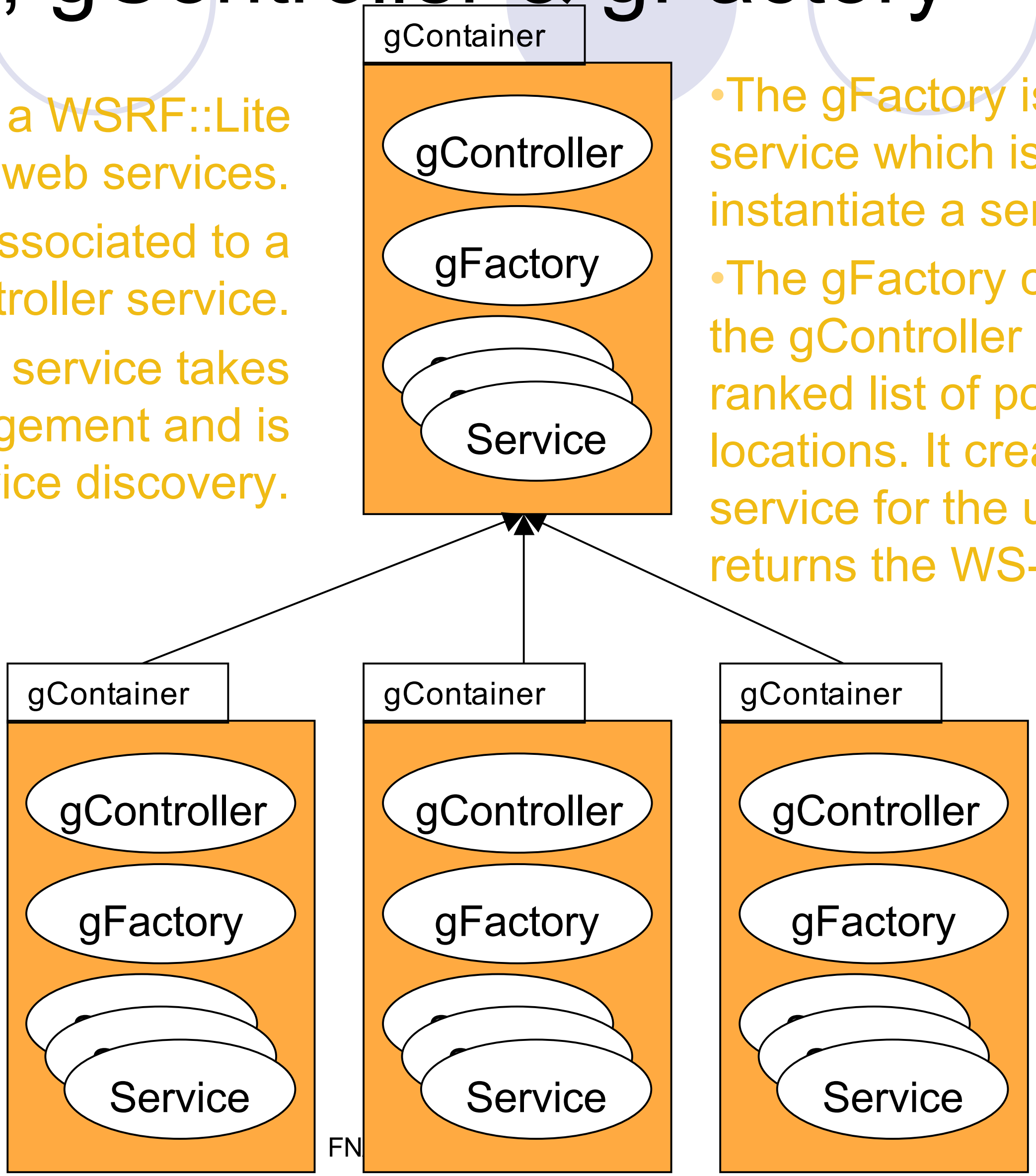
Interface snippets identical to interfaces of underlying services



gContainer, gController & gFactory

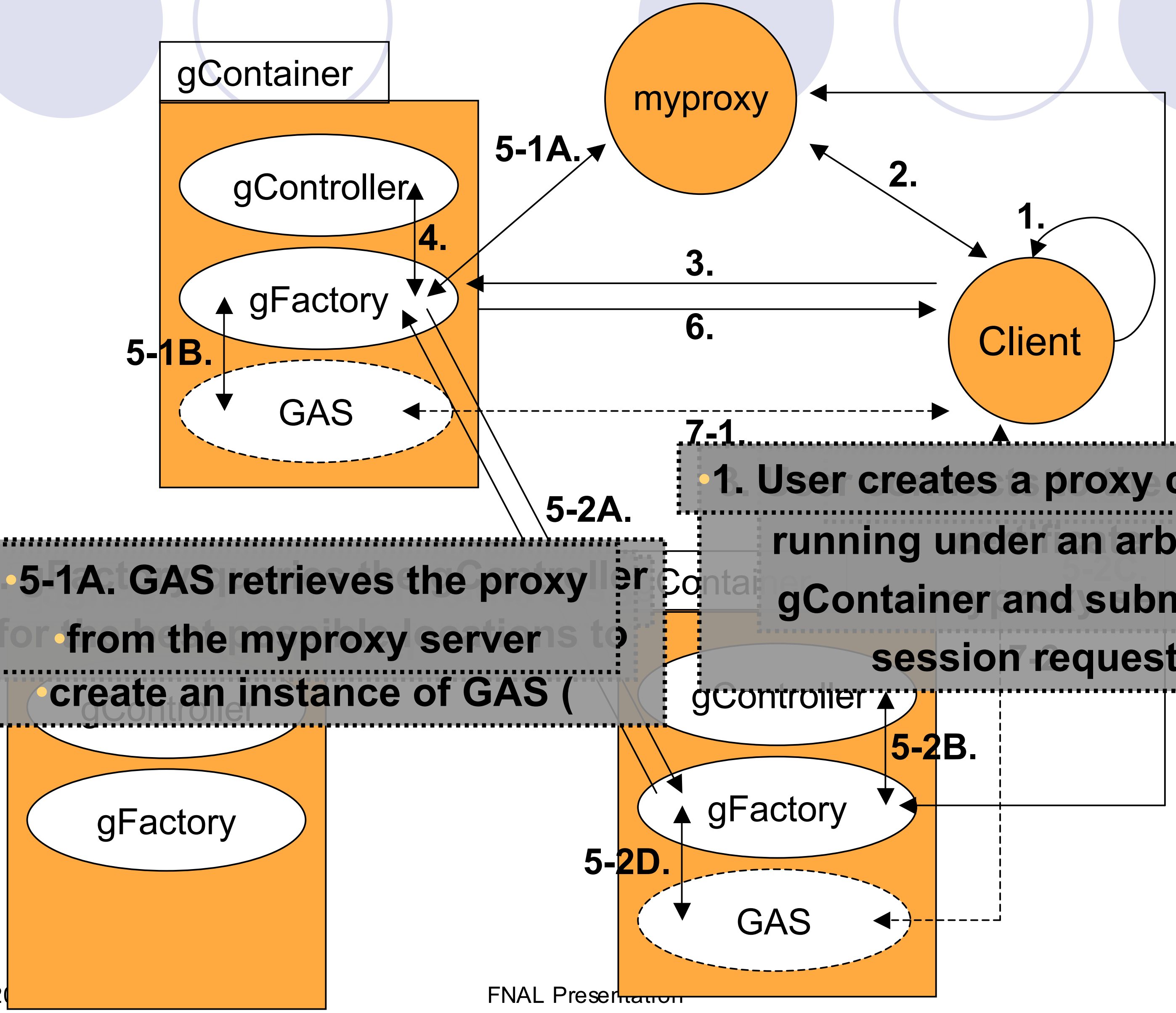
- The gContainer is a WSRF::Lite container which hosts web services.
- Every gContainer is associated to a gController service.
- gController stateful service takes care of load management and is used for service discovery.

- The gFactory is a stateless service which is used to instantiate a service (GAS).
- The gFactory connects to the gController to get a ranked list of possible locations. It creates the service for the user and returns the WS-Address.



- Every gContainer regularly contacts its parent and advertises its capabilities and the capabilities of its children. .

GAS, gContainer & myproxy



- 5-1A. GAS retrieves the proxy from the myproxy server
- create an instance of GAS (

- 1. User creates a proxy certificate

running under an arbitrary gContainer and submits a session request

GAS Status

- Core (nearly) complete
  - Authentication (myproxy; no renewal yet)
  - Discovery, Creation and Lifetime Management (integrated in gContainer)
- Interfaces
  - FileCatalog defined
  - MetaCatalog defined but will probably change (ARDA)
  - WMS under construction
- Integrated services
  - AliEn File Catalog
  - AliEn Meta Catalog
  - Java Meta Catalog for biomedical application

Outlook

- All experiment had to “complement” the existing Grids
- Someone developed services “on top”, someone an alternative stack
- The tendency is to virtualise services, so that we do not have different stacks, but rather different services that can be run on different underlying grid architectures
- The direction we are exploring is to exploit as much as possible the underlying grid architecture and “complement” them with our own services
  - Lightweight
  - In user space